Kodowanie i bezpieczeństwo informacji

Laboratorium - lista nr 4 na 26 III - 5 IV 2012

Podpisy cyfrowe nie są stosowane jedynie jako zamiennik podpisu odręcznego wykonywanego przez osobę. Podpisuje się np. pliki multimedialne zakupione w sklepach typu iTunes, sterowniki urządzeń w Windows, instalowane pakiety/aplikacje (RPM/Apt/Yum/..., Android, iPhone), dane uaktualniające program antywirusowy...

Twoim zadaniem (1, 2) jest zaimplementowanie silnika kryptograficznego tworzącego i weryfikującego podpisy pod plikami w jednym z powyższych scenariuszy. Każde z zadań można zrobić na co najmniej dwa sposoby – np. w zadaniu 2 można albo "zaszyć" certyfikat CA i werfikować podpis wraz z certyfikatami pośrednimi albo "zaszyć" jeden klucz publiczny; w zadaniu 1 można podpis przechowywać w osobnym pliku albo dopisywać do pliku oryginalnego albo przechowywać plik wraz z podpisem w jednym pliku wykorzystując format XaDeS. Zastanów się jakie są dobre i złe strony każdego z rozwiązań (w zależności od scenariusza).

W każdym z zadań wykorzystaj bibliotekę BouncyCastle (`http://www.bouncycastle.org/`) jako providera operacji kryptograficznych. Dzięki takiemu podejściu będziesz w stanie łatwo zmodyfikować program, aby w przyszłości wykorzystywał np. karty chipowe czy akceleratory kryptograficzne.

Zadanie 1 (4 pkt) Korzystając z kluczy, które zostały wygenerowane na potrzeby poprzedniej listy napisz aplikację podpisującą wskazany plik.

Zadanie 2 (3 pkt) Napisz aplikację, która uruchomi wskazany program jedynie gdy ten będzie podpisany odpowiednim kluczem prywatnym.

Zadanie 3 (3 pkt) Napisz aplikację, która wyświetla dane z bazy (baza danych może być np. plikiem xml, SQLite/MySQL/...). Identyfikatory w tej bazie są jawne, natomiast odpowiadające im pola są zaszyfrowane kluczem k (wykorzystaj AES w jednym z bezpiecznych trybów szyfrowania, vide: NIST SP 800-38E/NIST SP 800-38a). Klucz ten aplikacja odczytuje z pliku (z keystore'a) w momencie jej uruchomienia. Keystore zawierający klucz musi być zabezpieczony hasłem.

indeks	ocena
123456	$AES_k(27)$
123123	$AES_k(22)$